Database Programming 2

Home Assignment

The main aim of this assignment is to have a Python application that loads JSON files related to products into a DBMS. The DBMS will take care of parsing the loaded data and convert it to a normalised data.

Question 1

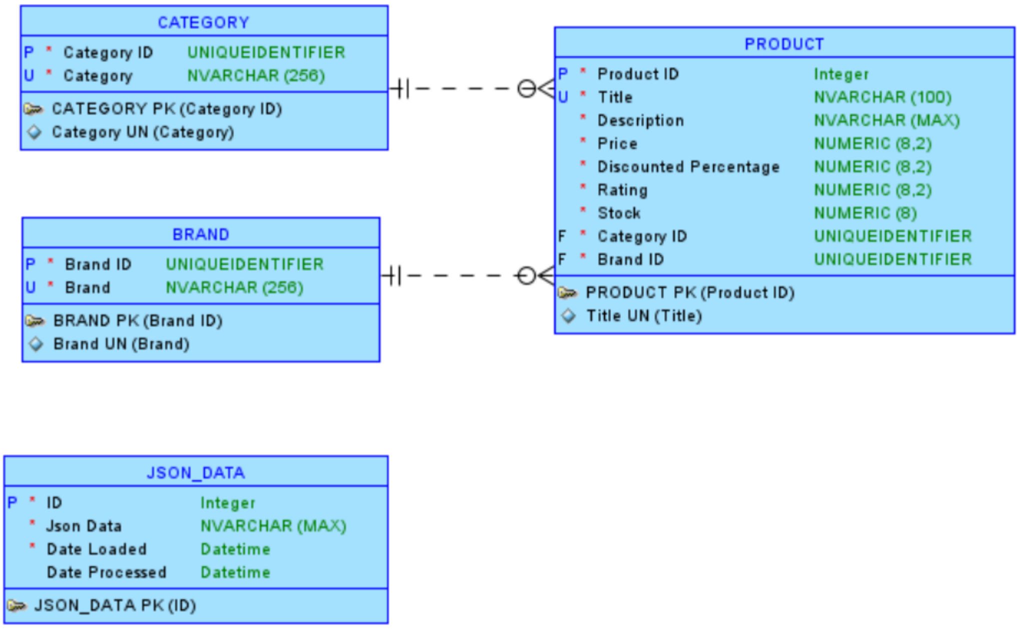
This part is setting up the required structured:

a) Create a database called *a3products.*

b) Create two schemas: *loading*, and *main*.

c) In *loading* schema, create the table *json_data* as per given ERD.

d) In *main* schema, create the tables *brand*, *category*, and *product* as per given ERD.



Question 2

Create a procedure named *loading.replaceQuotes* that accepts an integer as an input parameter. This procedure will update the loading.json_data table for the row with the specified id (passed as a parameter). The *json data* field should have all its single quotes replaced with double quotes.

Question 3

Create a procedure *main.processJson* that accepts an integer. This procedure must have the following functionality. You can see the data in attached files *products_1.json* and *products_2.json*.

a) Turn off row counting.

b) Declare a temporary table that will store only the following fields from the JSON data: *id*, *title*, *description*, *price*, *discountedPercentage*, *rating*, *stock*, *brand*, *category*.

c) Find the record from *loading.json_data* that has an ID that matches the passed parameter, process the JSON data and store it in the temporary table.

d) Insert the unique brands obtained from the temporary table into the table *main.brand*. Make sure not to insert existing brands.

e) Insert the unique categories obtained from the temporary table into the table *main.category*. Make sure not to insert existing categories.

f) Insert the products obtained from the temporary table into the table *main.product*. Make sure not to insert existing products (you can check this using the productid).

Question 4

Create a trigger on the table *loading.json_data* that will fire after a record has been inserted. The trigger must have the following functionalities:

a) Start a transaction.

b) Set the isolation level to prevent dirty reads and nonrepeatable reads.

c) Get the ID of the inserted record.

d) Use the ID obtained in part C and try to replace the quotes using the previously created procedure *loading.replaceQuotes*. This must have error handling. If not successful throw an error message with number 60001, state 1 and an appropriate message; then rollback the transaction.

e) Use the ID obtained in part C and try to process the JSON data using the previously created procedure *main.processJson*. This must have error handling. If not successful throw an error message with number 60002, state 1 and an appropriate message; then rollback the transaction.

f) If part D and E were successful, update the *date processed* field in the *loading.json_data* table with the current time and commit the transaction.

Question 5

Create a function *main.getProductsRating* that accepts a rating decimal number, and it returns the *title*, *price*, *rating*, *stock*, and *brand* for all products having a rating greater or equal to the passed parameter. The return result must be formatted in JSON as shown below:

```
[
  {
    "title": "cereals muesli fruit nuts",
    "price": 46,
    "rating": 4.94,
    "stock": 113,
    "brand": "fauji"
  },
  {
    "title": "Key Holder",
    "price": 30,
    "rating": 4.92,
    "stock": 54,
    "brand": "Golden"
  },
…
]
```

Question 6

Create a Python application with the following requirements:

a) A function called *createConnection* that accepts a server name, and a database name. The function should establish a connection to the DBMS and return the connection. Hint: pyodbc package can be used.

b) A function called *closeConnection* that accepts a database connection as a parameter, it closes the connection, and exits the program.

c) A global variable called *conn* where the database connection is stored.

d) A function called *loadData* that accepts a file path. This function must have the following functionality:

   a. Loads the JSON data from the given file (parameter). Hint: you can use *json* package.

   b. Insert the JSON data into the *loading.json_data* table. Hint: the JSON data must be converted to string.

   c. You must use error handling in this function taking care of 3 types of errors as follows. For each case, make sure to print an appropriate message.

      i. An error that is returned from pyodbc.

      ii. An error that is returns if the file is not found.

      iii. A generic error.

e) A function called *getRatings* that accepts a decimal as a parameter. The function must have the following functionality:

   a. It must call the function *main.getProductsRating*.

   b. It must print the result of the function as a table. Hint: you can use *pandas* package to read JSON data.

   c. You must use error handling in this function taking care of 2 types of errors as follows. For each case, make sure to print an appropriate message.

      i. An error that is returned from pyodbc.

      ii. A generic error.

f) A function called *showMenu* that displays the following options. The menu must keep appearing until the users chooses to exit.

   a. Option 1: Load Data – this will ask the use for a file location, and then call the function *loadData*.

   b. Option 2: Get Products by Rating – this will ask for a rating from 1 to 5, and then call *getRatings* function.

   c. Option 3: Exit – this will call *closeConnection*.

Marking Scheme

SE1.5 Design and develop a complete database solution.		
	Maximum	Awarded
Question 1a – Database created correctly.	0.5	
Question 1b – Correct schemas created.	0.5	
Question 1c/d – Tables created correctly. (For any mistake deduct 1 mark)	4	
Question 2 – Procedure to replace quotes works correctly.	3	
Question 3 – processJson procedure created correctly. (Row counting 0.5 marks).	2	
Total	10	

KU4.1 Show how to query JSON data within a DB		
	Maximum	Awarded
Question 3b – Temporary table created correctly.	2	
Question 3c – Process to insert json data into loading.json_data is correct.	3	
Total	5	

SE3.3 Evaluate a database system and improve its resilience.		
	Maximum	Awarded
Question 3d – Table brand filled in correctly.	2	
Question 3e – Table category filled in correctly.	2	
Question 3f - Table product filled in correctly.	3	
Question 4c – ID retrieved correctly.	1	
Question 4f – Time updated correctly.	2	
Total	10	

SE3.4 Develop a series of queries that cater for concurrency issues.		
	Maximum	Awarded
Question 4 – Trigger signature is correct.	1	
Question 4a/b – Transaction is set correctly.	2	
Question 4a/b – Commit and Rollback correctly used.	2	
Question 4d – replaceQuotes correctly called with error handling	2.5	
Question 4e – processJson correctly called with error handling	2.5	
Total	10	

AA4.3 Prepare several reports generated from JSON data stored within a DB.		
	Maximum	Awarded
Question 5 – getProductsRating works correctly.	4	
Question 6 – JSON data correctly presented as a table	3	
Total	7	

AA4.2 Develop an application with CRUD functionalities that communicates with a DB.		
Question 6	Maximum	Awarded
createConnection developed correctly.	1	
closeConnection developed correctly.	0.5	
Global variable for connection used.	0.5	
loadData works successfully.	2	
getRatings works successfully.	2	
Menu works successfully.	1	
Total	7	